

STI 3^{ème} année – Programmation Système

TD 1: Norme POSIX

C. Toinard – J.-F. Lalande - J. Briffaut

1 Environnement

Les variables d'environnement sont des variables dynamiques utilisées par les différents processus d'un système d'exploitation

- L'environnement est accessible via : **extern char **environ;**
- Ou par la fonction POSIX :

```
#include <stdlib.h>
char *getenv(const char *nomDeVariable);
int putenv(const char *coupleNomValeur);
```

Exercice 1 Tapez "env" dans la console. Vous obtenez la liste de toutes les variables d'environnement du système.

Exercice 2 Ecrire un programme C équivalent à la commande env.

Exercice 3 Ecrire un programme C qui prend en argument le nom d'une variable d'environnement et affiche sa valeur.

2 Appel système et gestion des erreurs

Exercice 4 Ecrivez un programme permettant d'écrire le message hello sur la sortie standard à l'aide d'un appel système.

Exercice 5 Modifiez ce programme afin qu'il génère une erreur lors de l'appel système.

Exercice 6 Traitez l'erreur retournée au moyen de perror (1 ligne).

Exercice 7 Sans utiliser perror, traitez l'erreur retournée au moyen de errno et de strerror. Il s'agit d'écrire un équivalent de la fonction perror qui écrit le texte explicatif suivi d'un message spécifique à l'application.

3 Accès aux limites du système

Les limites du systèmes ne dépendent pas forcément que du système. Dans le cas des fichiers par exemple, une limite peut dépendre du fichier que l'on considère.

Exercice 8 Etudiez la fonction posix pathconf. Ecrivez une fonction **void pr_pathconf(char *path, int name, char * message)** qui prend en paramètre le nom de fichier à considérer, l'nom de la limite (entier) et le message textuel à afficher pour cette limite. La fonction envoie sur la sortie standard le nom de la limite puis appelle la fonction pathconf

afin de récupérer la valeur de la limite. La fonction envoie alors sur la sortie standard la valeur récupérée. Traitez les cas d'erreur avec `errno` de façon rudimentaire afin d'afficher le même message signalant l'erreur à la place du nom de la limite et de sa valeur.

Exercice 9 Créez un programme `main` permettant de tester votre fonctions. Votre `main` ne testera que les limites POSIX.1.

4 Base de données utilisateur

Exercice 10 Ecrivez un programme qui prend en paramètre un nom d'utilisateur et affiche le contenu de l'enregistrement correspondant de la base de données des comptes.

Exercice 11 Proposez une méthode pour le tester pour l'ensemble des comptes existants sur votre système.

Exercice 12 Comment se fait-il que vous ne trouviez pas votre login dans les utilisateurs systèmes alors que votre programme est capable d'afficher les informations relatives à votre login ?